UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/820,661	04/07/2004	Paul A. Martin	SUN04-0234	8024

57960 7590 06/24/2009
PVF -- SUN MICROSYSTEMS INC.
C/O PARK, VAUGHAN & FLEMING LLP
2820 FIFTH STREET
DAVIS, CA 95618-7759

EXAMINER	
KIM, PAUL	

ART UNIT	PAPER NUMBER
2169	

MAIL DATE	DELIVERY MODE
06/24/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/820,661
Filing Date: April 07, 2004
Appellant(s): MARTIN, PAUL A.

Anthony P. Jones
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 24 March 2009 appealing from the Office action mailed 3 September 2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

McGregor et al, Practical C++, published by Que, 11 August 1999

7,225,210	Guthrie	11-2003

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. **Claims 1, 4, 6-9, 11, 13-14, 15, 18, 20-23, 25, 27-28, 29, 32, 34-37, 39, and 41-42** are rejected under 35 U.S.C. 103(a) as being unpatentable over McGregor (NPL, "Practical C++, published by Que on 11 August 1999) in view of Guthrie, II (U.S. Patent No. 7,225,210, hereinafter referred to as Guthrie), filed on 20 November 2003, published on 26 May 2005, and issued on 29 May 2007, and in further view of Thatte et al (U.S. Patent No. 4,853,842, hereinafter referred to as Thatte), filed on 19 February 1988, and issued on 1 August 1989.

3. **As per independent claims 1, 15, and 29,** McGregor, in combination with Guthrie and Thatte, discloses:

> A method for performing a lock-free update to one or more fields in an existing node in a linked list, comprising:
>
> receiving a reference to the existing node in the linked list, wherein the existing node contains the one or more fields to be updated {See McGregor, page 8, wherein this reads over "start at either the head or the tail and traverse the list until the item is found"};
>
> obtaining a new node to be added to the linked list, wherein other processes do not possess references to the new node and therefore cannot initially access the new node {See McGregor, page 9, wherein this reads over "allocated a new node"};
>
> copying a snapshot of the existing node to the new node {See Guthrie, col. 8, lines 20-67, wherein this reads over "the snapshot system created a new node 6 and incremented the snapshot identifier of node 0 to 1"}, which includes copying a next pointer of the existing node to the new node, so that the new node points to a node immediately following the existing node {See McGregor, page 7, wherein this reads over "the address of the new node is assigned to a node pointer"; and page 11, wherein this reads over "all you have to do is swap the pointers of the nodes on either side of the node you are deleting"};

updating one or more fields in the new node that correspond to the one or more fields in the existing node {See McGregor, page 9, wherein this reads over "set the integer value for the node"};

performing a single atomic operation that modifies the next pointer of the existing node to point to the new node and also marks the next pointer to indicate that the existing node is deleted, whereby the new node becomes part of the linked list and the existing node is deleted in a single atomic operation {See McGregor, page 11, wherein this reads over "all you have to do is swap the pointers of the nodes on either side of the node you are deleting"}; and

splicing the existing node out of the linked list by atomically modifying the next pointer of a node immediately preceding the existing node in the linked list to point to the new node, instead of pointing to the existing node {See McGregor, page 11, wherein this reads over "all you have to do is swap the pointers of the nodes on either side of the node you are deleting"}

wherein copying the snapshot of the existing node to the new node further involves examining the next pointer of the existing node to determine if the existing node has been deleted {See Thatte et al, C9:L29-59, wherein this reads over "no outstanding pointers to the object from non-garbage objects" and "the outstanding pointers to the object may point to empty space, i.e., unallocated memory, or to some undesired object" and "[a] deleted object is specially marked as tombstoned when it is deleted}.

While McGregor may fail to expressly disclose the copying of a snapshot of an existing node, Guthrie discloses a snapshot data system wherein snapshots of nodes comprising a plurality of fields are taken and copied. Accordingly, the combination of invention disclosed by McGregor and Guthrie would disclose an invention which would comprise a method for obtaining a snapshot of an existing node and applying said copy to a newly created node such that updates may be made to the newly created node without necessitating a lock of the existing node. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the above invention suggested by McGregor by combining it with the invention disclosed by Guthrie.

One of ordinary skill in the art would have been motivated to do this modification such that a newly created node, which consists of copied fields which have been updated, may be inserted into a linked list.

While McGregor may fail to expressly disclose the method step of "wherein copying the snapshot of the existing node to the new node further involves examining the next pointer of the existing node to determine if the existing node has been deleted," Thatte discloses a method wherein objects may be specially marked as tombstoned and any outstanding pointers may be checked. Accordingly, the combination of inventions disclosed by McGregor and Thatte would discloses an invention wherein the existing node may be tombstoned and all related pointers of said existing node may be deleted in a garbage collection process. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the above invention suggested by McGregor by combining it with the invention disclosed by Thatte.

One of ordinary skill in the art would have been motivated to do this modification such that it may be determined prior to copy whether an existing node has already been deleted.

4. **As per dependent claims 4, 18, and 32,** McGregor, in combination with Guthrie and Thatte, discloses:

> The method of claim 1, wherein copying a snapshot of the existing node to the new node involves:
>
> copying the contents of the existing node to the new node {See Guthrie, col. 8, lines 43-54, wherein this reads over "[t]he snapshot system copied the data of root node 0 to the root node 6 of the snapshot"};
>
> examining the next pointer of the existing node to determine if the existing node has been deleted {See McGregor, page 8, wherein this reads over "CIntList::Find( ) function"}; and
>
> if so taking a remedial action;
>
> otherwise, not taking the remedial action.

5. **As per dependent claims 6, 20, and 34,** McGregor, in combination with Guthrie and Thatte, discloses:

> The method of claim 1, further comprising deleting a target node from the linked list by:
>
> receiving a reference to the target node to be deleted from the linked list {See McGregor, page 8, wherein this reads over "CIntList::Find( ) function"};
>
> atomically marking a next pointer in the target node to indicate that the target node is deleted {See McGregor, page 11, wherein this reads over "all you have to do is swap the pointers of the nodes on either side of the node you are deleting"}; and
>
> atomically modifying the next pointer of a node immediately preceding the target node in the linked list to point to a node immediately following the target node in the linked list, instead of pointing to the target node, thereby splicing the target

node out of the linked list {See McGregor, page 11, wherein this reads over "all you have to do is swap the pointers of the nodes on either side of the node you are deleting"}.

6. **As per dependent claims 7, 21, and 35,** McGregor, in combination with Guthrie and Thatte, discloses:

The method of claim 6, wherein after the target node is spliced out of the linked list, the method further comprises modifying the next pointer of the target node so that the next pointer remains marked but points to a node immediately preceding the target node instead of the node immediately following node the target node in the linked list {See McGregor, page 11, wherein this reads over "all you have to do is swap the pointers of the nodes on either side of the node you are deleting"}.

7. **As per dependent claims 8, 22, and 36,** McGregor, in combination with Guthrie and Thatte, discloses:

The method of claim 1, further comprising inserting an additional node into the linked list by:

identifying a node immediately preceding the additional node in the linked list {See McGregor, page 8, wherein this reads over "CIntList::Find( ) function";

identifying a node immediately following the additional node in the linked list {See McGregor, page 8, wherein this reads over "CIntList::Find( ) function"; and

splicing the additional node into the linked list by, setting the next pointer for the additional node to point to the immediately following node {See McGregor, page 11, wherein this reads over "all you have to do is swap the pointers of the nodes on either side of the node you are deleting"}, and

atomically updating the next pointer of the immediately preceding node to point to the additional node {See McGregor, page 11, wherein this reads over "all you have to do is swap the pointers of the nodes on either side of the node you are deleting"}.

8. **As per dependent claims 9, 23, and 37,** McGregor, in combination with Guthrie and Thatte, discloses:

The method of claim 1, further comprising

reading a snapshot of multiple fields from a target node in the linked list by:

reading the multiple fields from the target node {See Guthrie, col. 8, lines 20-38, wherein this reads over "[t]emplate 100 illustrates the fields of the node"};

examining the next pointer of the target node to determine if the target node has been deleted {See McGregor, page 8, wherein this reads over "CIntList::Find( ) function"}; and if so, taking a remedial action;

otherwise, not taking the remedial action.

9. **As per dependent claims 11, 25, and 39,** McGregor, in combination with Guthrie and Thatte, discloses:

The method of claim 1, wherein atomically modifying the next pointer of the existing node to indicate that the existing node is deleted involves setting a "deleted bit" in the next pointer {See McGregor, page 11, wherein this reads over "reset the node's address to zero"}.

10. **As per dependent claims 13, 27, and 41,** McGregor, in combination with Guthrie and Thatte, discloses:

The method of claim 1, wherein a given node in the linked list includes:

a key that contains an identifier for the given node {See McGregor, page 4, wherein this reads over "Find ( ) Returns a pointer to the first node containing the specified integer value"};

one or more fields containing data values or pointers to data values associated with the given node {See Guthrie, col. 8, lines 20-38, wherein this reads over "[t]emplate 100 illustrates the fields of the node"}; and

a next pointer that contains the address of a node that immediately follows the given node in the linked list, and that also contains a deleted indicator, which indicates whether the given node has been deleted {See McGregor, page 7, wherein this reads over "the address of the new node is assigned to a node pointer"; and page 11, wherein this reads over "all you have to do is swap the pointers of the nodes on either side of the node you are deleting"}.

11. **As per dependent claims 14, 28, and 42,** McGregor, in combination with Guthrie and Thatte, discloses:

The method of claim 1, further comprising periodically performing a garbage-collection operation to reclaim deleted nodes that have become unreachable {See McGregor, page 11, wherein this reads over "[d]elete the node (which has now been unlinked from the list), reset the node's address to zero"}.

12. **Claims 3, 17, and 31** are rejected under 35 U.S.C. 103(a) as being unpatentable over McGregor, in view of Guthrie and Thatte, and further in view of Official Notice.

13. **As per dependent claims 3, 17, and 31,** the Examiner takes Official Notice that it would have been obvious and widely-known to one of ordinary skill in the art to execute a splicing process should a prior process fail to perform the splicing operation.

14. **Claims 5, 10, 12, 19, 24, 26, 33, 38, and 40** are rejected under 35 U.S.C. 103(a) as being unpatentable over McGregor, in view of Guthrie and Thatte, and further in view of Lippman et al (NPL, "C++ Primer," published by Addison Wesley Professional on 2 April 1998.

15. **As per dependent claims 5, 19, and 33,** McGregor, in combination with Guthrie, Thatte, and Lippman, discloses:

The method of claim 4, wherein taking the remedial action involves:

> following the next pointer of the existing node in an attempt to find an updated version of the existing node {See McGregor, page 8, wherein this reads over "CIntList::Find( ) function"};
>
> if an updated version of the existing node is found, copying a snapshot of the updated version of the existing node to the new node {See Guthrie, col. 8, line 55 – col. 9, line 9, wherein this reads over "[t]he snapshot system then created a new node for the node being modified" and "set the snapshot identifier field of node 8 to 2 and set the previous field of node 8 to 2"}; and
>
> if an updated version of the existing node is not found, indicating that the remedial action fails {See Lippman, page 5, wherein this reads over "assert ( )"}.

While McGregor and Guthrie may fail to expressly disclose a method of providing an indication of a remedial action failure, Lippman discloses a method for announcing a condition that triggers the assertion. Accordingly, the combination of invention disclosed by McGregor, Guthrie, and Lippman would disclose an invention which would comprise a method for announcing a triggering condition. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the above invention suggested by McGregor, Guthrie, and Thatte by combining it with the invention disclosed by Lippman.

One of ordinary skill in the art would have been motivated to do this modification such that an indication may be provided that a certain remedial action failed.

16. **As per dependent claims 10, 24, and 38,** McGregor, in combination with Guthrie, Thatte, and Lippman, discloses:

The method of claim 9, wherein taking the remedial action involves:

> following the next pointer of the target node in an attempt to find an updated version of the target node {See McGregor, page 8, wherein this reads over "CIntList::Find( ) function"}; and
>
> if an updated version of the target node is found, repeating the process of reading a snapshot of the multiple fields from the updated version of the target node {See Guthrie, col. 8, line 55 – col. 9, line 9, wherein this reads over "[t]he snapshot system then created a new node for the node being modified" and "set the snapshot identifier field of node 8 to 2 and set the previous field of node 8 to 2"}; and
>
> if an updated version of the existing node is not found, indicating that the remedial action fails {See Lippman, page 5, wherein this reads over "assert ( )"}.

While McGregor and Guthrie may fail to expressly disclose a method of providing an indication of a remedial action failure, Lippman discloses a method for announcing a condition that triggers the assertion. Accordingly, the combination of invention disclosed by McGregor, Guthrie, Thatte, and Lippman would disclose an invention which would comprise a method for announcing a triggering condition. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the above invention suggested by McGregor, Guthrie, and Thatte by combining it with the invention disclosed by Lippman.

One of ordinary skill in the art would have been motivated to do this modification such that an indication may be provided that a certain remedial action failed.

17. **As per dependent claims 12, 26, and 40,** McGregor, in combination with Guthrie, Thatte, and Lippman, discloses:

> The method of claim 1, wherein while atomically modifying the next pointer of the existing node,
>
> if the next pointer indicates that the existing node is already deleted, the atomic modification operation fails and the method further comprises taking a remedial action to deal with the fact that the existing node is already deleted {See Lippman, page 5, wherein this reads over "assert ( )"};
>
> otherwise, continuing performing the atomic modification operation.

While McGregor and Guthrie may fail to expressly disclose a method of providing an indication of a remedial action failure, Lippman discloses a method for announcing a condition that triggers the assertion. Accordingly, the combination of invention disclosed by McGregor, Guthrie, Thatte, and

Lippman would disclose an invention which would comprise a method for announcing a triggering condition. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the above invention suggested by McGregor, Guthrie, and Thattte by combining it with the invention disclosed by Lippman.

One of ordinary skill in the art would have been motivated to do this modification such that an indication may be provided that a certain remedial action failed.

(10) Response to Argument

Rejections of Independent Claims 1, 15, and 29

Appellant asserts the argument that "Examiner has failed to explain how McGregor's disclosure of a two-step node-deletion operation renders obvious the present invention's single atomic operation for updating a node in the linked list." See Appeal Brief, page 16. The Examiner respectfully disagrees. For the purposes of examination, the Examiner notes that, within the art, an "atomic operation" refers to a set of operations that can be combined so that they appear to the rest of the system to be a single operation with the outcome being of either a success or a failure. Accordingly, the Examiner notes that it would have been obvious to one of ordinary skill in the art that the operative steps as disclosed by McGregor would be combinable to produce a single atomic operation.

a. McGregor Discloses a Single Atomic Operation

Appellant asserts the argument that "the pointer swapping operation for deleting a node is a two-step operation which includes two separate sequential steps." See Appeal Brief, page 17. The Examiner respectfully disagrees. It is noted that Appellant's claimed invention also provides an operation comprising a two-step operation which includes two separate sequential steps: (1) inserting a new node into the linked list; and (2) marking an existing node to indicate removal. Accordingly, it is unclear to the Examiner how Appellant's "two-step" atomic operation may be distinguished from McGregor's two-step operation. That is, wherein the "delete" and "insert" operations of McGregor would appear to the rest of the system to be a single operation with the outcome being either a success or a failure, by definition, said operations may be considered to be "atomic operations". Accordingly, the Examiner notes that McGregor does not teach directly away from the single atomic operation claimed by Appellant as the sub-operations of the delete and insert operations are formed to provide single delete and insert operations.

b. McGregor Discloses a Node-Deletion Operation Fundamentally Similar to the Update Operation of the Present Invention

Appellant asserts the argument that "the two-step operation in McGregor does not insert a new node into a linked list, whereas the single-atomic operation in the instant application inserts a new node into a linked list." See Appeal Brief, page 19. The Examiner respectfully disagrees. It is noted that McGregor discloses an insertion operation wherein a new node is allocated (i.e. "[t]he first step is to allocate a new node"). See McGregor, page 9. Wherein McGregor further discloses that the pointers of a node may be assigned new addresses, it would have been obvious to one of ordinary skill in the art that a new node may indeed be inserted into the linked list.

Secondly, Appellant asserts that "McGregor removes a node from the linked list, whereas the single-atomic operation in the instant application marks an outdated version node to indicate impending deletion." See Appeal Brief, page 19. The Examiner respectfully disagrees. Wherein Appellant has failed to indicate and recite how one may mark "the next pointer to indicate that the existing node is deleted," the Examiner construes said recitation such that the actual deletion of the pointer to the existing node would "indicate that the existing node is deleted." That is, wherein the pointer to an existing node is deleted and no longer available, said existing node would not be accessible such that it would be expressly indicative of the existing node's deletion.

Lastly, Appellant asserts the argument that "Guthrie doe not suggest or imply performing a single atomic operation to achieve the two described objectives." See Appeal Brief, page 19. The Examiner notes that Guthrie was used to disclose the feature of copying a snapshot of an existing node. It is noted that the two described objectives are instead disclosed by McGregor for the aforementioned reasons above.

For the aforementioned reasons above in subparagraphs (a) and (b), the claim rejections under 35 U.S.C. 103 are maintained.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Paul Kim/

Paul Kim
Patent Examiner, Art Unit 2169

Conferees:
Neveen Abel-Jalil signing for Tony Mahmoudi , SPE 2169
/Neveen Abel-Jalil/

Supervisory Patent Examiner, Art Unit 2165

/Mohammad Ali/

Supervisory Patent Examiner, Art Unit 2158